Int. J. on Recent Trends in Engineering and Technology, Vol. 8, No. 1, Jan 2013

A Novel Method for Keyword Retrieval using Weighted Standard Deviation: "D4 Algorithm"

Divya D Dev[1], G.S. Anandha Mala[2], P. Muthuchelvi[3]
[1]Computer Science and Engineering, St. Joseph's College of Engineering, OMR Chennai 119, India
Email: divyaddev@gmail.com
[2, 3]St. Joseph's College of Engineering, OMR Chennai 119, India
Email : [2]gs.anandhamala@gmail.com, [3]muthuchelvi69@gmail.com

Abstract - **Genetic Algorithm (GA) has been a successful method that is been used for extracting keywords. This paper presents a full method by which keywords can be derived from the various corpuses. We have built equations that exploit the structure of the documents from which the keywords need to be extracted. The procedures are been broken into two distinguished profiles: one is to weigh the words in the whole document content and the other is to explore the possibilities of the occurrence of key terms by using genetic algorithm. The basic equations of the heuristic mechanism is been varied to allow the complete exploitation of document. The Genetic Algorithm and the enhanced standard deviation method is used in full potential to enable the generation of the key terms that describe the given text document. The new technique has an enhanced performance and better time complexities**.

Index Terms – **Genetic Algorithms, Weighted Term Standard Deviation, TF*IDF, D4 Algorithm**

I. Introduction

Research showed that digitalisation had gained massive esteem in recent years [4]. Text classification techniques required an enhanced feature selection method that allowed the topic to be extracted from the plain textual method [12]. The techniques relied on exploiting the various documental parameters like the number of words in the document, the position of the words and the number of word occurrences. The keyword extraction had been used in text classifications and summarisation [2].

In this paper, the keyword extraction was done by using the basic equation stated by the Weighted Term Standard Deviation method in an extended fashion. The basic weighing functions were being followed by the genetic procedures. The use of the existing methodologies together with the newly devised content was proposed to the end of the paper. The research aimed in navigating through all the document parameters. The focus of the work improved the output generated and could be further used in the classification. Each document comprised of a topic. The documents had an appropriate title that was used to describe the content. The title was followed by the body of the document. The body being not a single entity, but a collection of many factors. The factors namely included the sub paragraphs that elaborated on the topic, links and other references.

The importance of the data varied according to the location of the content within the document. The initial words of a paragraph had been given quite more priority than the others that occurred later in the document. The main attempt in the extraction of the keywords would be to use the maximum level of information from the document structure in collaboration with the words. The extraction of the words avoided repeating a given word in the obtained set of key terms. Such perfection could have been gained only when the corpus was dealt off fully considering all its variation methods. The word weighing was built based on the above information that included the reputation given to the individual words and the whole document itself.

In the extraction, the proper representation of the document was vital. The existing methods worked on the certain aspects of the document features and by which the word profile was established. The features and its weights were then passed onto the genetic algorithm procedures. The total number of words derived was based on the dimensionality of the content. One of the profile methods dealt with the conversion of the document to a feasible form from which the weights of the words would be calibrated. Then the genetic algorithm is included to enhance the feature weighing schemes.

Several test documents had been used to decide the efficiency of the extraction. The end result was compared against a prebuilt set of words and their weights. The similarity comparison values were also been computed to ensure the output. The commonly used extraction method was the term frequency and the inverse document frequency. The prominence of the term in a given document had been captured by the term frequency and the segregation of the term across the document was stated by the inverse document frequency. The contribution of this method worked towards the distinguishing of terms based on the number of occurrences.

The methods of term frequency had failed on several occasions. Various researches showed that the terms had been given weights on their relevance frequency method [7]. Apart from the above, the engineering of the feature words in machine learning methods also required rules to be defined manually. The efficiency was not reliable and impractical. In this paper, the technique was exploited that didn't rely on manpower and varied the outputs across all possibilities of the text parameters. It ensured the accuracy and optimisation of the end results.

The paper is organised as follows: Section 2: details on the literature of feature extraction. Section 3: based on the design and development of the new method for keyword

DOI: 01.IJRTET.8.1.27

extraction and introduced the experimental procedures for the performance measures. Section 4: deals with the genetic algorithm methods. The Section 5: analyses the data and the results obtained. Section 6: concludes with the summary, identifying the limitations of the research and proposed suggestions for better future study.

II. Related Work

The weighing of the feature words was broken down into two parts based on the impact of the given terms across the whole collection. The fundamental was the term frequency (TF) where the occurrences of the term in the document were considered [7]. The inverse document frequency (IDF) gave the weighing to the whole range of documents and therefore the equation was being compiled in to the following:

$$tfidf\ (ti, dj) = tf\ * \log(\ N\ /\ n) \quad (1)$$

Where tf - Term frequency of the i^{th} term in j^{th} document
N- Total number of document in the collection
n - Document frequency of i^{th} term

As stated previously the term was deemed to serve to the net output generated with respect to the number of times it appeared in the document. Where the positive consideration was given when the occurrence was more and less priority was given for the ones that occurred less frequently. The relevance frequency (RF) [11][6] was given by the following:

$$RF(t) = \log(1 + n(+) / n(-)) \quad (2)$$

Where n(+)- number of positive documents with the term
n(-)- number of negative documents with the terms

Feature selection methods like the Keyword Extraction Algorithm (KEA) were generated later on. The new KEA++ include the phrase length of words and the degree of various nodes resultant from which the weights get computed, this was absent in the KEA's initial versions. Various factors like the document type, the structure of the corpus and the context were used to help the feature calculation [1]. The existing techniques were not reliantly fast when the above considerations were made. Berend et al., [6] gave the feature extraction at corpus and knowledge based level. Yet the preceding involved the reliance on external knowledge sources, to ensure the performance. Similarly Wan et al., [9] enabled the derivation of the single document keywords by using the supporting document sets.

An affinity graph was being built using the key words that were filtered by various syntactic measures. The ranking was done by the graph based ranking algorithm that identified each word. Ahead of the others techniques would be the supervised and unsupervised methods.

The unsupervised models had been groomed recently and were way more achievable in terms of performance when compared to the supervised natures [8]. The supervised involved the classification of the key phrase during the training phase. The KEA came under the supervised strategy. The other supervised methodologies seen by Zhang et al., [11] included the Support Vector Machine (SVM) procedures. In spite of the output relevance the major fall of the supervised methods included the requirement of corpus annotation and the extraction of unknown keyphrases.

The supervised framework as proposed in [3] has included a concept which goes beyond the traditional frequency measures. The literature included the linguistically specific, decision making sentence features. It had a level of motivation towards term specific attributes and prosodic prominence scores. The supervised framework evolved with a feedback loop which showed a clear summary between the keywords and the sentences. Nevertheless the technique proposed in [3] did fail in certain ways due to the frequency oriented approach. The human notations and evaluation issues delivered an echelon of errors.

Position oriented keyword extraction methods have taken into consideration only the words which occurred in special positions. This includes the headline, the title, the first paragraph and the most important sentences [15]. The preceded work focused with an idea that the keywords lay only in the important sentences. These assumptions had to be eliminated for prominent results. Taking into consideration the above missed out factors, the paper proposes a new technique for keyword extraction, namely the D4 Algorithm. The D4 Algorithm evolved around four principle computational techniques. The first phase deals with the measure of the "Density". This is where each of the individual terms in the document is assigned with a unique weight. Following which the "Double cross" is done. This is the application of procedures by which the suitable combination of parent terms is being identified. Thereafter the "Depth" indentifies whether the terms lay within the fitness threshold, based on which the "Devour" else "Mutation" is carried out to identify the final weight of the word.

III. D4 Algorithm

A. Preparing the Document

The text documents were given as a direct input into the algorithm. The content needed to be converted to the appropriate form for the processing [5]. Therefore the procedures for the stop word removal and stemming were applied to the sample text. The stemming involved the words being converted to their basic representation. The punctuation marks were removed along with the numbers. All the words were converted to their lower case representation. The stop words were removed by comparison with an already existent list of stop words. The advantage of the document preprocessing included the increase of the dimensionality. The space complexity was reduced along with a decrease in the processing of additional words. An assumption made was that words with just two characters would not contribute to the meaning thus the terms fitting to the above category had been eliminated.

B. Overview of the D4 Algorithm

The overall processes involved in the D4 Algorithm, feature word extraction is given below. In which the initial

DOI: 01.IJRTET.8.1.27

few blocks showed the tasks of the "initial weighing" followed by the phases of "genetic algorithm". The initial weighing standards allow the system to assign an individual weight to each individual word. The characterisation of the term occurs through numerical values.

The weighted term outputs are subject to the divide and conquer technique. This broke the sorted word list into two sets of terms. Thereafter the application of genetic algorithm began. This was when the stages of crossover identified the two suitable parents; from the divided weighted word lists and then the simple probability crossover equations were applied. Following this, the fitness of the terms was being checked. If the terms have not attained stability they would have being subject to mutation.

The Fig. 1 gives an outlook of the D4 algorithm. The D4 algorithm runs extensively on genetic procedures until the fitness is been fulfilled or the termination criteria is been met. The equations used in the crossover, fitness and mutation are devised to frame an effectual keyword extraction mechanism. The flow was reliant on the weight of the words. The changes occurred in such a way that the final result lay within a small state space.

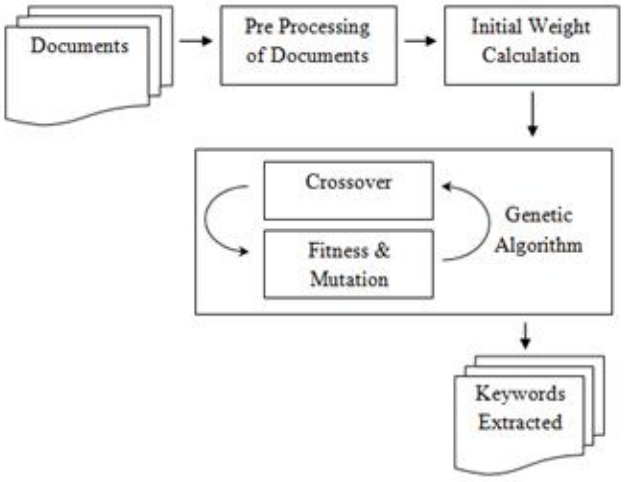

Figure 1. The flow of control in the D4 Algorithm

C. Initial Weight Computation of the word

The information retrieval systems all relied on the weight of the words. As far as our method was considered the whole process worked with the weight of the terms.

In the D4 algorithm, the term weighing was broken into two parts. The first part focused on the initial weighing of the words that acted as the representation of the terms. The second part involved the application of the genetic engineering modules. The equations used in the genetic algorithm allows the generation of the terms that represent the content. The processes worked such that words with a lower weight were of no significance to the documental values. The initial part of weighing the terms comprised of two segments. When the words occurred for the first time the representation in equation (3) was used. This was an extension of the weighted term standard deviation method as given by [13]. The equation that was proposed previously exploited only certain document features thus reducing the accuracy. However the new formulation weighed the terms based on their presence with respect to the individual sentences and the whole document.

The equation (3) was applied when a word appears for the foremost time. It considers both the document and sentence parameters. The weights calculated for each word by combining equations (3) and (4) had shown to possess extra precision than the existing techniques.

$$W(t_i) = \frac{Scount}{Tpos} * \frac{Maxline}{D_{\max}} \tag{3}$$

Where

W(ti) - Initial Weight of the word in its first occurrence

Tpos - Position of word in the sentence

Scount - Total number of words in the sentence

D_{max} - Maximum number of words in the document

Maxline - Count of the words that is in the line which has the maximum number of terms in the document

Loc - Average position of word in the document

When the words started repeating the next profile equation (4) was used. This created a base in which the terms that appeared many times had a greater priority in the document. The formula at time "t" used the weight that was computed for the word at time "t-1" and this was continued till the last occurrence of the term in the given document. The equation now deviated from the base equations and surfed through the salient sentences and unique document features. This increased the accuracy and relevance of the output.

$$W(ti) = \log(\Pr(W(ti))) + \sqrt{\left|Tpos - (Tpos^2)/D_{\max}\right| loc} \tag{4}$$

The dimensionality reduction helped the evaluation. Words with the lower weights gave a reduced prominence to the document's meaning. Words that occur deeper down in the passage seemed to have less weightage but the positional factors of the words with respect to the particular sentence neutralised the drop. If a term appears in the very first line, and repeats itself again in the last line. The formulation works in such a way that "just because the word happened to be in the first sentence it cannot be a keyword as it did repeat itself again in the last sentence (second appearance)".

A balance was made between the position of the word in the document and of that in the given sentence. The combination of the equations (3) and (4) gave an enhanced weighted term standard deviation measure that produced a better set of keywords, to be passed on to the next phase of Genetic Algorithms. The balance described above acts as the strength of the D4 algorithm's first phase. The following Algorithm 1 gives an overview of the initial weight computation.

The results of the initial weighing equations were passed onto the genetic procedures. The weights of the terms were sorted in ascending order before the methods of crossover, fitness and mutation was applied. The real number weights had several advantages over the binary encoding method. The optimization was being attained by this "no conversion method". The population was passed without the duplication

DOI: 01.IJRTET.8.1.27

Algorithm 1 for calculating the initial weights of the words

Let S_n be the sentences in the document in extraction phase
Let n = ϕ be the number of sentences in S
Let c = ϕ be the words in a sentence S
Let w = ϕ be the set of words in document
Let d = ϕ be the distinct words in document
Let wt = ϕ be the weight of the words in the document
for i from 1 to n **do**
 for j from 1 to c **do**
  if sentence s_i contains the word w_j that is not in d **then**
   Add w_i to d
   Calculate wt_i of word w_i according to (3)
else
   Calculate wt_i of word w_i according to (4)
  End if
 End for
End for

of words. The probability crossover and mutation values were chosen so that premature convergence did not occur [10][14].

The crossover probability was evaluated using the merging of the most possible word weights with the low feasible word weights. The survival of the fittest was ensured. The chances were given to the words which had a lower level of feasibility. The words that did not stand to the survival rates would be eliminated by further exploitation of the given equations [14].

D. Crossover

The equations were in such a way that the population that has lower chances of survival was subject to sufficient amounts of modulation in order to check for any chances of future survival. The methodologies behind the crossover involved the divide and conquer technique. The weighted term set was sorted in descending order. The population was broken into two halves. The first half had the highly feasible word set, which carried the population of terms that owned the higher weights and the second comprised the lower weighted terms. The equations exploited in the crossover mechanism included the following:

$$r(i) = \frac{X(i)}{X(i) + X(k)} 0 \le i \le Doc(mid) \quad (5)$$

$$r(k) = \frac{X(k)}{X(i) + X(k)} Doc(mid) \le k \le Doc(\max) \quad (6)$$

$$P(i) = (r(i) * X(i)) + ((1 - r(k)) * X(k)) \quad (7)$$

$$P(k) = (r(k) * X(k)) + ((1 - r(i)) * X(i)) \quad (8)$$

Where
X(i) - Weight of the word in position "i"
P(k) - the probability crossover parameter of the word in position "k".

The P(i) and P(k) had the probability crossover of the word that used both sides of the population. This was where the D4 algorithm exploits both the most feasible words with the lower practicable ones. The probability method of "p = (1-q)" was used to give the likeliness of one word with respect to another. The "p" gave the weight of the highly prioritised word and "q" represented the lower priority one. The above calibration had shown chances of the least feasible terms to move up the rank listings. The foundation of the crossover was to achieve a population by choosing a suitable cross breed of parents. These parents had to have the ability to generate a crowd that could satisfy the fitness calculations. The values of the crossover changed the implementation of the whole algorithm. The unhealthy values of the crossover could lead to an un-optimal local solution that would get trapped in a diverse state space. The diverse situation could lead to the termination [16] and the cessation of the algorithm. Fig. 2 demonstrates the breaking down of the sorted weighted terms into two segments. This is referred to be the divide and conquer stage. The pairing was done with respect to value of "n". The combination was different when the "n" is odd or even. When the value of n is odd there would be a single term that goes unpaired. This weight of this word will not be subject to any alterations through crossover equations. Nevertheless the midterm would keep changing in the different iterations, as the weight values of the other terms were modified.

Fig. 3 shows the application of the double crossover expressions. The paired parents were passed into the crossover equations (5) to (8). After the crossover equations, the word set were rearranged and passed into the phase of fitness evaluation.

The robustness of the solution depended on the selection of parents passed from the computation done at time "t" to the next iteration namely "t+1". The fitness function had been used to produce an objective functional value of relative fitness expressions. The survival of an individual term was considered separately without taking the erstwhile

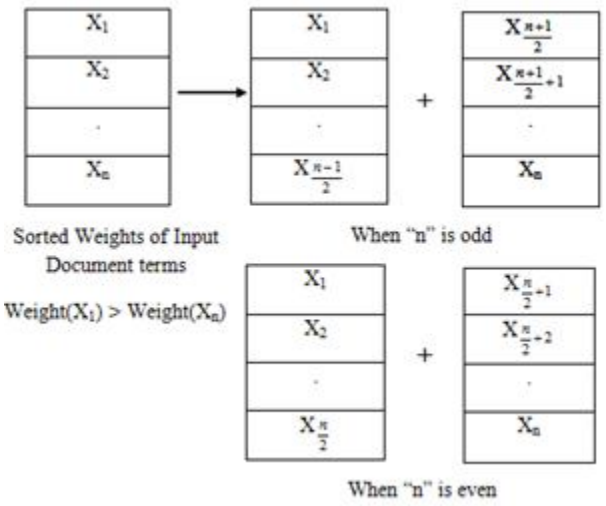

Figure 2. Phase 1 of crossover – Divide and Conquer

E. Fitness Equation

The algorithm circulated with the principle of "Expected number of copies of the word "i" in the forthcoming population". The equational representation would be as (9). Avg(P) was the Average Probability Crossover. The value of the Avg(P) is equivalent to the subsequent equation (11).

DOI: 01.IJRTET.8.1.27

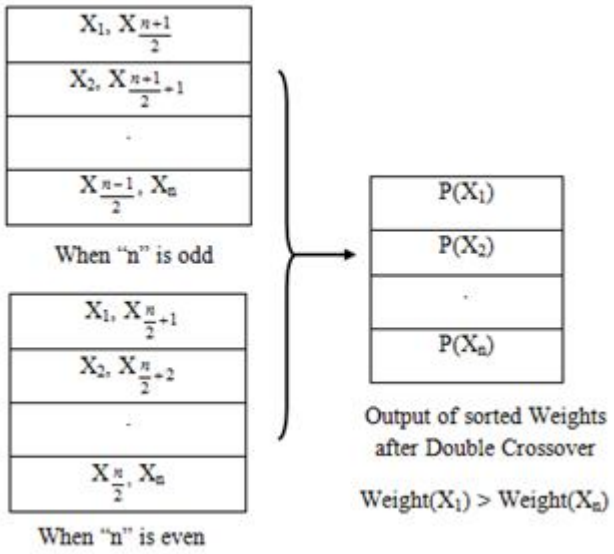

Figure 3. Phase 2 of Crossover – Application of Equations

$$E(n_i) = \frac{P(i)}{Avg(P)} * D_{\max} \quad (9)$$

$$T(i) = E(n_i) * D_{distinct\text{-}terms\text{-}in\text{-}document} \quad (10)$$

$$Avg(P) = \frac{\sum_{i=0}^{n} P(i)}{n} \quad (11)$$

Mutation involved in the modification of the weight of the word in the solution, obtained from the fitness function with a probability mutation of "pm". The mutation had the ability to restore the lost genetic material into the population thus preventing the convergence of the solution into a sub-optimal one. The decision of whether the word should be mutated or not was given by T(i) in equation (10). The words below the fitness threshold were subject to mutation. The probability of the mutation was dependent on the fitness value. When further iteration did not change the value of the weights, and all the words lay well above the fitness threshold, the stable state was attained. Henceforth the ultimate keyword list could be generated to describe the document. The enhanced algorithm did not mutate the fit terms. The words were mutated independently from the neighbours in the given state space. The mutation was depicted by the equational representation given below. Where M(i) denoted the muted weight of terms.

$$T(i) \le Avg(P) \Rightarrow M(i) = \sqrt{P(i)} + \left(found(i) / Avg(Position(i)) \right) \quad (12)$$

The words that were not fit had a new weight computed i.e. M(i). The mutation equation was not applicable if the weight of a term lay in the fitness equation. The M(i) replaced the existing weight of a word. The weights of the muted and the steadfast words will be then combined to form the new population. This was passed again into the genetic computation of crossover. Once all the terms came within the fitness threshold the genetic algorithm terminates.

IV. Performance Evaluation

A. Reuters 21578

Three different measures were used for the study namely the percentage of keywords extracted, the F Measure and the Entropy. The test documents were the Reuter-21578, manually processed set of abstracts and the Test data set of KEA. The Reuter-21578 had been widely used for the testing of keyword extraction mechanisms. The Reuters-21578 was an experimental data collection that appeared on the Reuters newswire of the year 1987. The dataset was obtained from http:kdd.ics.uci.edu/databases /reuters21578 /reuters21578. It contained nearly 8000 documents classified under different topics. This paper focused on comparing the new method with the existent weighted term standard deviation [14], the discourse method [13] against the outputs gained through the TD*IDF method. Several authors had shown that the TF*IDF method was an antique and consistent result producer over large time spans. Hence the results of the TF were been used as the baseline for our evaluation. It was found that the output of the novel method surpassed the existing ones. Table 1 show the percentage of keywords extracted by the individual extraction techniques. The F-Measure was being used for the evaluation of performance. The Table 2 shows the F-Measure of the computation. The values show that the output of the D4 Algorithm performed marginally better the previous functionalities. The results did justify that the new way of extraction was more appropriate than the existent ones. Entropy measure could also be used for the homogenous evaluation. The total entropy was evaluated as the sum of the precision. It was necessary to minimize the entropy of the results. The Entropy in Table 3 proved our proposed method to be better than the previous extraction technique.

Table I. Percentage of keywords extracted from reuters 21758

Number of keywords	% Keywords Extracted		
	WTSD	Discourse	D4
3	84.7066	52.2066	94.5233
4	76.1055	67.9959	78.0707
5	54.5988	96.1113	54.5988
6	58.1816	47.4814	63.0224
7	50.3285	18.8178	52.3528
8	50.8537	28.8028	50.4675
9	44.6767	61.1344	61.8733

Table II. F-measure of the extraction in reuters 21758

Number of keywords	F-Measure		
	WTSD	Discourse	D4
3	0.2718	0.0625	0.3076
4	0.2325	0.0575	0.2406
5	0.2765	0.0835	0.2765
6	0.1784	0.1545	0.2055
7	0.1937	0.1707	0.2079
8	0.1611	0.1348	0.1585
9	0.0700	0.0693	0.1618

DOI: 01.IJRTET.8.1.27

TABLE III. ENTROPY OF THE EXTRACTION IN REUTERS 21758

Number of keywords	Entropy		
	WTSD	Discourse	D4
3	0.28357	0.57464	0.25412
4	0.31228	0.58827	0.30442
5	0.27299	0.52023	0.27299
6	0.37812	0.40683	0.34909
7	0.36647	0.39146	0.35230
8	0.40374	0.43763	0.40683
9	0.55686	0.55854	0.40203

B. Abstracts

The testing was also extended to the data set that was being retrieved from the following link ftp://ftp.cs.cornell.edu/pub/smart . This contained around 500 documents out of which 423 were perfect, with proper text content. The Table 4 gives the output in terms of the number of keywords obtained for the above dataset. The result shows that the D4 was better than the existent techniques. The overhead was seen prominently between the discourse and the new method. However in the abstract evaluation the weighted term standard deviation was appealingly parallel to the proposed algorithm.

C. KEA

KEA has a standard keyword list attached with every document. The dataset was pre-processed and executed under the WTSD, Discourse and D4 Algorithm. The keywords that were extracted by these methods were compared against the standard keyword list attached with each document. The Table 5 gives an outlay of the performance delivered by the various extraction methods.

TABLE IV. PERCENTAGE OF KEYWORDS EXTRACTED FROM ABSTRACTS

Number of keywords	% Keywords Extracted		
	WTSD	Discourse	D4
3	56.66	10.00	66.67
4	63.00	12.33	66.00
5	99.86	23.00	99.36
6	64.40	54.00	76.00
7	80.00	69.67	87.00
8	73.33	60.33	72.01
9	33.33	33.33	83.33

TABLE V. PERCENTAGE OF KEYWORDS EXTRACTED FROM KEA

Document Category	% Keywords Extracted		
	WTSD	Discourse	D4
1	56.66	20.00	66.67
2	63.00	10.00	66.00
3	99.86	1.000	99.36
4	64.40	12.50	76.00
5	80.00	14.28	87.00
6	73.33	11.11	72.01
7	33.33	20.00	83.33

V. CONCLUSION

In this study the D4 algorithm proposed a feature selection method with the weighted term standard deviation as its baseline. The D4 Algorithm has approached to use the conventional term weighing methods followed by the application of the genetic algorithm. The new D4 Algorithm had an increased precision and the recall performances. The above testing tables clearly demonstrated that the new feature selection method was much better than the existing ones. The feature selection method was not limited by the size of the database. The accuracy was retained and explored even when large data sets had been used. The net output was better than the existing technique and it extracted the right number of keywords which varies from one document to another. The technique over took the previous techniques as the survival of the fittest was always been taken into consideration. The GA gave a better hand in finding the words describing the document. The initial weight equation showed the application of all the possible document attributes. This included the number of words in a document, the total number of sentences, the words in each sentence, the number of distinct words, the average position of the words in the document and the number of times a term repeats. The D4 Algorithm worked to ensure maximum efficiency. There is room for improvement where the number of keywords extracted should be within a given state space that did not cover a larger part of the document size thus reducing the words extracted and concentrate the keywords on a list that would gain a better F-Measure.

REFERENCES

[1] Wei You, Dominique Fontaine, Jean-Paul Bathes, "An automatic keyphrase extraction system for scientific documents," *Knowledge Information System Springer-Verlag London*, DOI 10.1007/s10115-012-0480-2, April 2012.

[2] Bolin Ding, Bo Zhao, Cindy Xide Lin, Jiawei Han, Chengxiang Zhai, Ashok Srivastave and Nikung C. Og, "Efficient keyword-based search for top-K cells in text cube," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, pp. 1795-1810, December 2011.

[3] Fei Liu, Feifan Liu and Yang Liu, "A supervised framework for keyword extraction from meeting transcripts," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, pp. 538-548, March 2011.

[4] H.H. Kian, and M. Zahedi, "An efficient approach for keyword selection; improving accessibility of web contents by general search engines," *International Journal of Web & Semantic Technology*, vol. 2, pp. 81-90, October 2011.

[5] Shady, S. Fakhri, and K. Mohamed, "An efficient concept based mining model for enhanced text clustering," *IEEE Transaction on Knowledge and Data Engineering*, vol. 22, pp. 1360–1370, October 2010.

[6] G. Berend, and R. Farkas, "Feature engineering for keyphrase extraction," *SemEval '10 Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 186–189, July 2010.

[7] M. Meenakshi Sundaram, K. Lakshmi, and M. Saswati, "A negative category based approach for wikipedia document classification," *Int J. Knowledge Engineering and Data Mining*, vol. 1, pp. 84-97, April 2010.

DOI: 01.IJRTET.8.1.27

[8] Z. Elberrichi, L. Bellatreche, M. Simonet, and M. Malki, "Concept - based clustering of textual documents using SOM," *IEEE/ACS International Conference on Computer Systems and Applications*, pp. 156–163, April 2008.

[9] Xiaojun Wan, and Jianguo Xiao, "Single document keyphrase extraction using neighbourhood knowledge," *Proceedings of 23rd national conference on Artificial intelligence*, vol. 2, pp. 855–860, 2008.

[10] S.M. Kholessizadeh, R. Zaefarian, S.H. Nasseri and E. Ardil, "Genetic Mining: Using genetic algorithm for topic based on concept distribution word," *World Academy of Science, Engineering and Technology*, pp. 144–147, 2006.

[11] Kuo Zhang, Hui Xu, Jie Tang and Juanzi Li, "Keyword extraction using support vector machine," *Proceedings of the 7th International Conference on Advances in Web-Age Information Management Springer-Verlag*, vol. 4016, pp. 85–96, 2006.

[12] Man Lan, Chew-Lim Tan, Hwee-Boon Low and Sam-Yuan Sung, "A comparative study on term weighting schemes for text categorization," *IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 546–551, August 2005.

[13] Tatsuya Kawahara, Masahiro Hasegawa, Kazuya Shitaoka, Tasuku Kitade and Hiroaki Nanjo, "Automatic indexing of lecture presentations using unsupervised learning of presumed discourse Markers," *IEEE Transactions on Speech and Audio Processing*, vol. 12, pp 409-419, July 2004

[14] Sung-Shun Weng and Yu-Jen Lin, "A study on searching for similar documents based on multiple concepts and distribution of concepts," *Expert Systems with Applications*, vol. 25, pp. 355–368, October 2003.

[15] S. Marsili Libelli and P. Alba, "Adaptive mutation in genetic algorithms," *Soft computing Springer-Verlag*, vol. 4, pp. 76–80, 2000

[16] M. Srinivas and L.M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, pp. 656–667, April 1994.

DOI: 01.IJRTET.8.1.27